

Design of tree structures for efficient querying, R. G. Casey, *Communications of the ACM* 16, No. 9, 549–556 (September 1973). A standard information retrieval operation is to determine which records in a data collection satisfy a given query expressed in terms of data values. The process of locating the desired responses can be represented by a tree search model. This paper poses an optimization problem in the design of such trees to serve a well-specified application. The problem is academic in the sense that ordinarily the optimal tree cannot be implemented by means of practical techniques. On the other hand, it is potentially useful for the comparison it affords between observed performance and that of an intuitively attractive ideal search procedure. As a practical application of such a model, this paper considers the design of a novel tree search scheme based on a bit vector representation of data and shows that essentially the same algorithm can be used to design either an ideal search tree or a bit-vector tree. An experimental study of a small formatted file illustrates the concepts.

Distributions of queue lengths and waiting times in a loop with two-way traffic, A. G. Konheim and B. Meister, *Journal of Computer and System Sciences* 7, No. 5, 506–521 (October 1973). A loop transmission system consisting of a central station, N input terminals with infinite buffer, and a multiplexed channel is considered. Data flows from the terminals to the central station and from the central station to the terminals. The first type of data transfer is accorded priority over the second type. This necessitates intermediate buffering of data from the central station to the terminals. The distributions of the queue lengths at all terminals and of the virtual waiting times of data units are calculated. Two numerical examples are given.

Investigation into scheduling for an interactive computing system, H. A. Anderson, Jr., and R. G. Sargent, *IBM Journal of Research and Development* 18, No. 2, 125–137 (March 1974). This paper describes a statistical evaluation of the performance of the swap scheduling algorithm of an interactive computer system and investigation into foreground-background scheduling to improve system performance. Input traffic, computer service time demands, and system performance were statistically analyzed. Based on the results of these analyses performance enhancements for the system were determined and then evaluated through use of a validated simulation model.

Queueing analysis of a multiprogrammed computer system having a multilevel storage hierarchy, S. S. Lavenberg, *SIAM Journal on Computing* 2, No. 4, 232–252 (December 1973). We formulate a class of closed queueing network models which can be used to represent certain features of multiprogrammed computer systems having multilevel storage hierarchies. The resources which comprise the system are described by a network of interconnected multiserver stages where each stage can provide more than one type of service. The sequence of services required by a program executing in the system is described by a finite Markov chain over the service types. This description permits an explicit representation in the model of the data transfers which occur as determined by the data transfer rules and data paths in the hierarchy. The queueing discipline at each stage is nonpreemptive priority among the types of service provided by the stage, and first-come-served within a service type. We derive simple expressions relating the work rates for different stages and obtain simple upper bounds on work rates. These results are valid for general service time distributions. We then apply a model in this class to the analysis of a multiprogrammed three-level staging hierarchy. Under the assumption that all service time distributions are exponential, we numerically investigate the effects on system performance of different service priorities and of varying the program load parameters and level of multiprogramming.

Abstracts

Some distribution-free aspects of paging algorithm performance, P. A. Franaszek and T. J. Wagner, *ACM Journal* **21**, No. 1, 31–39 (January 1974). The topic of this paper is a probabilistic analysis of demand paging algorithms for storage hierarchies. Two aspects of algorithm performance are studied under the assumption that the sequence of page requests is statistically independent: the page fault probability for a fixed memory size and the variation of performance with memory. Performance bounds are obtained which are independent of the page request probabilities. It is shown that simple algorithms exist which yield fault probabilities close to optimal with only a modest increase in memory.

A technique for the solution of massive set covering problems, with application to airline crew scheduling, J. Rubin, *Transportation Science* **7**, No. 1, 34–48 (February 1973). Recent set covering algorithms have been able to solve problems for which the constraint matrix has as many as 10^4 columns. Unfortunately, in certain applications, the number of columns is combinatorially dependent on the number of rows, and can reach many orders of magnitude greater, for 500–1000 rows. For these problems, the constraint matrix cannot be generated, much less solved, unless severe ad hoc limitations are imposed. It seems clear that we must reluctantly abandon the search for the true mathematical optimum in such cases. One method of attack is to use a set covering algorithm repeatedly on much smaller matrices extracted from the overall problem, generating columns as needed. Such an approach has been used on an airline crew-scheduling problem, with excellent practical success on test cases involving close to 1000 rows. It utilizes some techniques that are more generally applicable, and some that make use of the structure of the crew-scheduling problem, and that therefore are specific to it.